# Hepatoprotective effects of systemic ER activation

ChIPseq/Epigenome genome - Enhancer-gene pair analysis

Christian Sommerauer & Carlos Gallardo

11 December, 2023

```
getwd()
```

```
## [1] "/Users/christiansom/Documents/GitHub/MAFLD_ER_agonists"
```

```
library(tidyverse)
```

## Use BEDOPS suite to determine the closest TSS to the enhancer sites. Run this in terminal, may adjust paths. Requires BEDOPS and bedtools.

```
#Import the closest genes as determined by
closest <- read.delim("results/Epigenome_analysis/Enhancer_Expression_correlation/origin_closest1_left_
enhancer_ID <- c(1:2181)

# Row 1, 451, 880, 1323, 2180 and 2181 have an NA in several columns (including V8), remove this one as
closest2 <- closest %>%
separate(col = V3, into = c("V3_left", "V3_right"), sep = "\\|") %>%
separate(col = V5, into = c("V5_left", "V5_right"), sep = "\\|") %>%
separate(col = V7, into = c("V7_left", "V7_right"), sep = "\\|") %>%
mutate(enhancer_ID=enhancer_ID) %>%
filter(!is.na(V8))

closest3 <- closest2 %>% mutate(enha.ident = paste0(V1, ".",V2,".",V3_left), .before = V1)

colnames(closest3) <- c("enha.ident", "ori_chrom", "ori_start", "ori_end", "left_1_chrom", "left_1_star
                        "right_1_chrom", "right_1_start", "right_1_end",
                        "right_2_chrom", "right_2_start", "right_2_end", "enhancer_ID")


  closest_ori <- closest3 %>% dplyr::select(1, 14, 2:4) %>% mutate(query_ID = "closest_ori")
  colnames(closest_ori) <- c("loc_ID", "enha.ID",  "chrom", "start", "end", "query_ID")
  closest_left1 <- closest3 %>% dplyr::select(1, 14, 5:7) %>% mutate(query_ID = "closest_left1")
  colnames(closest_left1) <- c("loc_ID", "enha.ID",  "chrom", "start", "end", "query_ID")

  closest_right1 <- closest3 %>% dplyr::select(1, 14, 8:10) %>% mutate(query_ID = "closest_right1")
```

```
  colnames(closest_right1) <- c("loc_ID", "enha.ID",  "chrom", "start", "end", "query_ID")
  closest_right2 <- closest3 %>% dplyr::select(1, 14, 11:13) %>% mutate(query_ID = "closest_right2")
  colnames(closest_right2) <-  c("loc_ID", "enha.ID",  "chrom", "start", "end", "query_ID")

closest_long <- rbind(closest_ori, closest_left1,
                      closest_right1, closest_right2)

#subset the enhancer_df to have the exact locations in three columns for later
location <- closest_ori %>% dplyr::select(1,3,4,5)
```

## Annotate the closest genes

```
library("ChIPpeakAnno")
library("GenomicRanges")
options(connectionObserver = NULL) #That is a work-around, as the org.Mm. package cannot be loaded
library("org.Mm.eg.db")
library("biomaRt")

gr_closest_long <- makeGRangesFromDataFrame(closest_long,  start.field = "start", end.field = "end", ig
names(gr_closest_long) <- c(1:length(gr_closest_long))
```

### Annotate the TSS

```
#listEnsemblArchives()
mart <- useMart(biomart = "ensembl", dataset = "mmusculus_gene_ensembl", host = "https://sep2019.archiv
annoDataMart <- getAnnotation(mart, featureType = "TSS")
```

### Annotate the TSS

```
gr_closest_long_anno <- annotatePeakInBatch(gr_closest_long,
                                AnnotationData=annoDataMart,
                                featureType = "TSS",
                                output="nearestLocation",
                                PeakLocForDistance = "start")

gr_closest_long_anno <- as.data.frame(gr_closest_long_anno)

table(gr_closest_long_anno$query_ID)
```

```
##
##  closest_left1    closest_ori closest_right1 closest_right2
##           2175           2175           2175           2176
```

## Import the gene expression data

```
getwd()
```

```
## [1] "/Users/christiansom/Documents/GitHub/MAFLD_ER_agonists"
```

```
source("code/00_helper_functions.R")
symbol_geneID <- read.delim("data/ensembl_mmus_sep2019_annotation.tsv")[,1:2]

raw_counts <- read.table(
  file = 'data/bulkRNAseq_mmus_rawcounts.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE) %>%
  dplyr::select(-PPT_HFD_male_4) %>%
  tibble::column_to_rownames('geneID') %>%
  as.matrix()

gene_len <- read.table(
  file = 'data/bulkRNAseq_mmus_gene_lengths.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE)

TPM <- normalizeData(x=raw_counts, len = gene_len$length, method = "TPM") %>%
  tibble::rownames_to_column("ensembl_gene_id")

TPM <- TPM %>%
  dplyr::select(ensembl_gene_id, CD_male_1,CD_male_3, CD_male_4, HFD_male_2,HFD_male_1,HFD_male_4, DPN_
TPM <- inner_join(symbol_geneID, TPM, by="ensembl_gene_id")

# We name the mice according to their original mouse number instead of replicate number of the RNAseq e
# CDm2, CDm6 and CDm9 correspond to CD1, CD3 and CD4
# HFDm3, HFDm4, HFDm6 correspond to HFD2, HFD1, and HFD4
# DPNm2, DPNm3 and DPNm6 correspond to DPN1, DPN2 and DPN3
# E2_2, E2_8 and E2_9 correspond to E2_2, E2_3 and E2_4
# PPTm1, PPTm2 and PPTm3 correspond to PPT1, PPT3 and PPT3

# DO NOT FORGET TO CHECK THAT HFD4 AND HFD3 ARENT MIXED UP
colnames(TPM) <- c("ensembl_gene_id", "symbol", "CDm2","CDm6", "CDm9", "HFDm3", "HFDm4","HFDm6","DPN2",

gr_closest_long_anno_closest_genes <- gr_closest_long_anno %>%
  filter(!query_ID=="closest_ori") %>%
  dplyr::rename("ensembl_gene_id"="feature")

gr_closest_long_anno_closest_genes <- as.data.frame(gr_closest_long_anno_closest_genes)

TPM_filt <- TPM %>%
  dplyr::filter(ensembl_gene_id%in%gr_closest_long_anno_closest_genes$ensembl_gene_id)

chrom_TPM <- inner_join(gr_closest_long_anno_closest_genes, TPM_filt, by= "ensembl_gene_id")
```

```
chrom_TPM2 <- chrom_TPM %>%
  dplyr::select("loc_ID" ,"seqnames", "start", "end", "enha.ID", "query_ID", "symbol", "ensembl_gene_id
```

# IMPORT ENHANCER COUNTS and normalize table

```
library(dplyr)
library(tidyr)

counts_enha <- read.delim("results/Epigenome_analysis/Reads_in_peaks_analysis/DAc_enhancers_2181_H3K27a
  dplyr::select("CK0744_H3K27ac_CD2", "CD6_H3K27ac_S2_R1_001", "CK0745_H3K27ac_CD9",
         "CK0746_H3K27ac_HFD3", "CK0747_H3K27ac_HFD4", "HFD6_H3K27ac_S8_R1_001",
         "CK0748_H3K27ac_DPN2", "CK0749_H3K27ac_DPN3", "DPN6_H3K27ac_S15_R1_001",
         "DIP3_H3K27ac_S9_R1_001",  "DIP6_H3K27ac_S10_R1_001", "DIP10_H3K27ac_S11_R1_001",
         "E2_2_H3K27ac_S16_R1_001", "CK0750_H3K27ac_E2_8", "CK0751_H3K27ac_E2_9",
         "PPT1_H3K27ac_S12_R1_001",  "PPT2_H3K27ac_S13_R1_001", "PPT3_H3K27ac_S14_R1_001")

names(counts_enha) <-  c("CDm2_K27","CDm6_K27", "CDm9_K27","HFDm3_K27","HFDm4_K27","HFDm6_K27", "DPN2_K

colsums_enha <- colSums(counts_enha[,])

counts_enha_norm <- sweep(counts_enha, 2, colsums_enha, FUN = "/")
counts_enha_norm2 <- counts_enha_norm *10^6
colSums(counts_enha_norm2[,])
```

```
##  CDm2_K27  CDm6_K27  CDm9_K27 HFDm3_K27 HFDm4_K27 HFDm6_K27  DPN2_K27  DPN3_K27
##     1e+06     1e+06     1e+06     1e+06     1e+06     1e+06     1e+06     1e+06
##  DPN6_K27  DIP3_K27  DIP6_K27 DIP10_K27  E2_2_K27  E2_8_K27  E2_9_K27  PPT1_K27
##     1e+06     1e+06     1e+06     1e+06     1e+06     1e+06     1e+06     1e+06
##  PPT2_K27  PPT3_K27
##     1e+06     1e+06
```

```
counts_enha_norm2.1 <- counts_enha_norm2 %>% rownames_to_column("loc_ID")
K27_GE_joined <- inner_join(chrom_TPM2, counts_enha_norm2.1, by="loc_ID")
#View(K27_GE_joined)

table(K27_GE_joined$query_ID)
```

```
##
##  closest_left1 closest_right1 closest_right2
##           2175           2175           2176
```

```
write.table(K27_GE_joined, "Supplementary_tables/SupplementaryTable_initial_ESEGs.txt", quote=F, row.nam
```

## Subset the enhancer table and put into long format

```r
sub_GE.K27_GE_joined <- K27_GE_joined %>%
  dplyr::select("loc_ID", "query_ID","symbol", "CDm2","CDm6", "CDm9", "HFDm3", "HFDm4","HFDm6","DPN2",
sub_GE.K27_GE_long <- pivot_longer(sub_GE.K27_GE_joined, cols=4:21, values_to = "Gene_expression")

sub.K27_K27_GE_joined <- K27_GE_joined %>%
  dplyr::select("loc_ID", "query_ID", "ensembl_gene_id","CDm2_K27","CDm6_K27", "CDm9_K27","HFDm3_K27","
sub.K27_K27_GE_long <- pivot_longer(sub.K27_K27_GE_joined, cols=4:21, values_to = "H3K27ac")

K27_GE_long <- cbind(sub_GE.K27_GE_long, sub.K27_K27_GE_long)
K27_GE_long_dd <- K27_GE_long[!duplicated(as.list(K27_GE_long))]

# Remove the zeros to not correlate zeros (gives error message - but these genes are removed later anyh
K27_GE_long_dd <- K27_GE_long_dd %>%
  group_by(loc_ID, symbol) %>%
  mutate(filter_zeros = mean(Gene_expression)) %>%
  filter(filter_zeros > 0) %>%
  dplyr::select(!filter_zeros)
```

## Import the reverted gene sets and filter the tables

```r
K27_GE_long_group <- K27_GE_long_dd %>% group_by(loc_ID, query_ID) %>%
  mutate(correlation_pearson = cor(Gene_expression, H3K27ac, method="pearson")) %>%
  mutate(corr.sig.pearson = cor.test(Gene_expression, H3K27ac, method="pearson")$p.value)

# Export a table for all ESEG with correlations BEFORE filtering anything.
write.table(K27_GE_long_group, "Supplementary_tables/SupplementaryTable_Initial_ESEG_genes_corr.txt", q

# Filter the ESEGs for a p.value > 0.01
K27_GE_long_group_plot_pearson <- K27_GE_long_group %>%
  filter(corr.sig.pearson < 0.01) %>%
  group_by(loc_ID, query_ID) %>%
  mutate(name.ident = paste0(symbol, "_", loc_ID))
nrow(K27_GE_long_group_plot_pearson)/18  # 589 enhancer gene pairs remain.
```

```
## [1] 589
```

```r
write.table(K27_GE_long_group_plot_pearson, "Supplementary_tables/SupplementaryTable_step2_ESEG_genes_c

# Import the reverted gene set (n=379)
DEGsets <- readRDS("results/bulkRNAseq_mmus_DEG_sets.rds")
revALL <- DEGsets$gene_id$reverted
length(revALL)
```

```
## [1] 379
```

```r
K27_GE_long_rev_insect <-  K27_GE_long_group_plot_pearson %>%
  filter(ensembl_gene_id %in% revALL)
length(unique(K27_GE_long_rev_insect$ensembl_gene_id)) # XXX unique genes remain after filtering for re
```

```
## [1] 67
```

```
nrow(K27_GE_long_rev_insect)/18 # 151 enhancer gene pairs remain after filtering
```

```
## [1] 151
```

```
write.table(K27_GE_long_rev_insect, "Supplementary_tables/SupplementaryTable_step3_ESEG_genes_corr_pval
```

#Add 50kb to intersect CTCF peaks with the H3K27ac peaks

```
K27_GE_long_group_coordinates <- inner_join(K27_GE_long_rev_insect, location, by="loc_ID")
K27_GE_long_group_coordinates$end <- as.integer(K27_GE_long_group_coordinates$end)

K27_GE_long_group_coordinates_left <- K27_GE_long_group_coordinates %>%
  dplyr::filter(query_ID=="closest_left1") %>%
  mutate(new_end = end+50000) %>%
  mutate(new_start=start)

K27_GE_long_group_coordinates_right <- K27_GE_long_group_coordinates %>%
  dplyr::filter(query_ID=="closest_right1" | query_ID=="closest_right2") %>%
  mutate(new_start = start-50000) %>%
  mutate(new_end=end)

K27_GE_long_group_coordinates_left_export <- K27_GE_long_group_coordinates_left %>%
  dplyr::select("chrom", "new_start", "new_end", "loc_ID", "query_ID", "symbol") %>%
  unique()

K27_GE_long_group_coordinates_right_export <- K27_GE_long_group_coordinates_right %>%
  dplyr::select("chrom", "new_start", "new_end", "loc_ID", "query_ID", "symbol") %>%
  unique()

write.table(K27_GE_long_group_coordinates_left_export, "results/Epigenome_analysis/Enhancer_Expression_
write.table(K27_GE_long_group_coordinates_right_export, "results/Epigenome_analysis/Enhancer_Expression

#From here, intersect the CTCF peaks with the exported H3K27ac regions using BEDtools (command line)
```

## prepare the CTCF files - separate by motif-orientation.

```
#load the motif-discovery file of CTCF motifs in mm10 genome by FIMO
FIMO_CTCF <- read.delim("data/fimo_mm10_genome_CTCFscan.tsv", sep="\t")

FIMO_CTCF_plus_bed <- FIMO_CTCF %>%
  dplyr::filter(strand=="+") %>%
  dplyr::select("chrom"="sequence_name", "start", "end"="stop", "strand")

FIMO_CTCF_minus_bed <- FIMO_CTCF %>%
  dplyr::filter(strand=="-") %>%
  dplyr::select("chrom"="sequence_name", "start", "end"="stop", "strand")

write.table(FIMO_CTCF_plus_bed, "results/Epigenome_analysis/Enhancer_Expression_correlation/fimo_mm10_g
write.table(FIMO_CTCF_minus_bed, "results/Epigenome_analysis/Enhancer_Expression_correlation/fimo_mm10_
```

HERE, RUN THE SHELL SCRIPT "Epigenome_06.03_CTCF_script_bedtools_enh_intersect.sh"

#after BEDTools intersection of H3K27ac enhancers (that have a good correlation with nearby genes) with nearby CTCF peaks, re-import

```
library(tidyverse)

names <- c("chrom", "start", "end", "loc_ID", "query_ID", "symbol")
H3K27ac_left_CTCFx_outwards <- read.delim("results/Epigenome_analysis/Enhancer_Expression_correlation/H3

H3K27ac_left_CTCFx <- read.delim("results/Epigenome_analysis/Enhancer_Expression_correlation/H3K27ac_le

H3K27ac_right_CTCFx_outwards <- read.delim("results/Epigenome_analysis/Enhancer_Expression_correlation/

H3K27ac_right_CTCFx <- read.delim("results/Epigenome_analysis/Enhancer_Expression_correlation/H3K27ac_r
```

#combine these data.frames, because they comprise the enhancer-gene pairs that we can report

```
H3K27ac_CTCF_intersect <- rbind(H3K27ac_left_CTCFx_outwards, H3K27ac_left_CTCFx, H3K27ac_right_CTCFx_ou
table(H3K27ac_CTCF_intersect$CTCF_pos)
```

```
##
## canonical  outwards
##        56        44
```

```
length(unique(H3K27ac_CTCF_intersect$loc_ID)) # Some enhancers (= loc_IDs) are duplicated due to multip
```

```
## [1] 76
```

```
unique_symbols <- unique(H3K27ac_CTCF_intersect$symbol)
length(unique_symbols)
```

```
## [1] 49
```

```
# 49 unique genes that may underlie enhancer-mediated estrogen-dependent regulation
```

#Compare the fold-change values for these sites - in addition to the reads in peaks this gives information about how much these enhancers are changed

```
CDvsHFD_H3K27ac_Diffbind <- readRDS("results/Epigenome_analysis/DiffBind_annotated_peaks/Diffbind_resul
 mutate(loc_ID = paste0(seqnames, ".",start,".",end), .before = seqnames) %>%
  dplyr::select(loc_ID, Fold, FDR) # Produce locIDs to match the ESEG IDs

# These are the enhancers intersected with CTCF. But more informative with foldchanges from Diffbind. T
H3K27ac_CTCF_intersect_log2FC <- inner_join(H3K27ac_CTCF_intersect,CDvsHFD_H3K27ac_Diffbind, by="loc_ID

# Retrieves the unique pearson corr values from BEFORE the CTCF intersection
corr_values <- K27_GE_long_rev_insect %>%
  ungroup() %>%
  dplyr::select("loc_ID", "correlation_pearson", "symbol") %>%
  unique() %>% group_by(symbol)
```

```r
# Creates a dataframe between the corr values and log2FCs.
H3K27ac_CTCF_intersect_log2FC_corr <- inner_join(corr_values, H3K27ac_CTCF_intersect_log2FC, by=c("loc_
duplicated(H3K27ac_CTCF_intersect_log2FC_corr)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE
```

```r
length(unique(H3K27ac_CTCF_intersect_log2FC_corr$symbol))
```

```
## [1] 49
```

```r
# This unique ID is necessary to join the data frames in the next section. Otherwise, enhancers that ar
H3K27ac_CTCF_intersect_log2FC_corr <- H3K27ac_CTCF_intersect_log2FC_corr %>%
  mutate(name.ident = paste0(loc_ID, ":", symbol))
```

```r
# To plot, we need the single columns for gene expression and log2FC again.  Note: some enhancers have
K27_GE_long_group_plot_filt <- K27_GE_long_rev_insect %>%
  group_by(symbol, loc_ID) %>%
  mutate(name.ident = paste0(loc_ID, ":", symbol)) %>%
  filter(name.ident%in%H3K27ac_CTCF_intersect_log2FC_corr$name.ident)
write.table(K27_GE_long_group_plot_filt, "Supplementary_tables/SupplementaryTable_step4_ESEG_genes_corr

K27_GE_long_group_plot_filt <- K27_GE_long_group_plot_filt[!duplicated(K27_GE_long_group_plot_filt), ]
length(unique(K27_GE_long_group_plot_filt$symbol))
```

```
## [1] 49
```

```r
# The following should yield "character(0)"
setdiff(K27_GE_long_group_plot_filt$symbol, H3K27ac_CTCF_intersect_log2FC_corr$symbol)
```

```
## character(0)
```

```r
length(unique(K27_GE_long_group_plot_filt$symbol)) # 49 unique genes
```

```
## [1] 49
```

```r
nrow(K27_GE_long_group_plot_filt)/18 # 80 enhancer - gene pairs
```

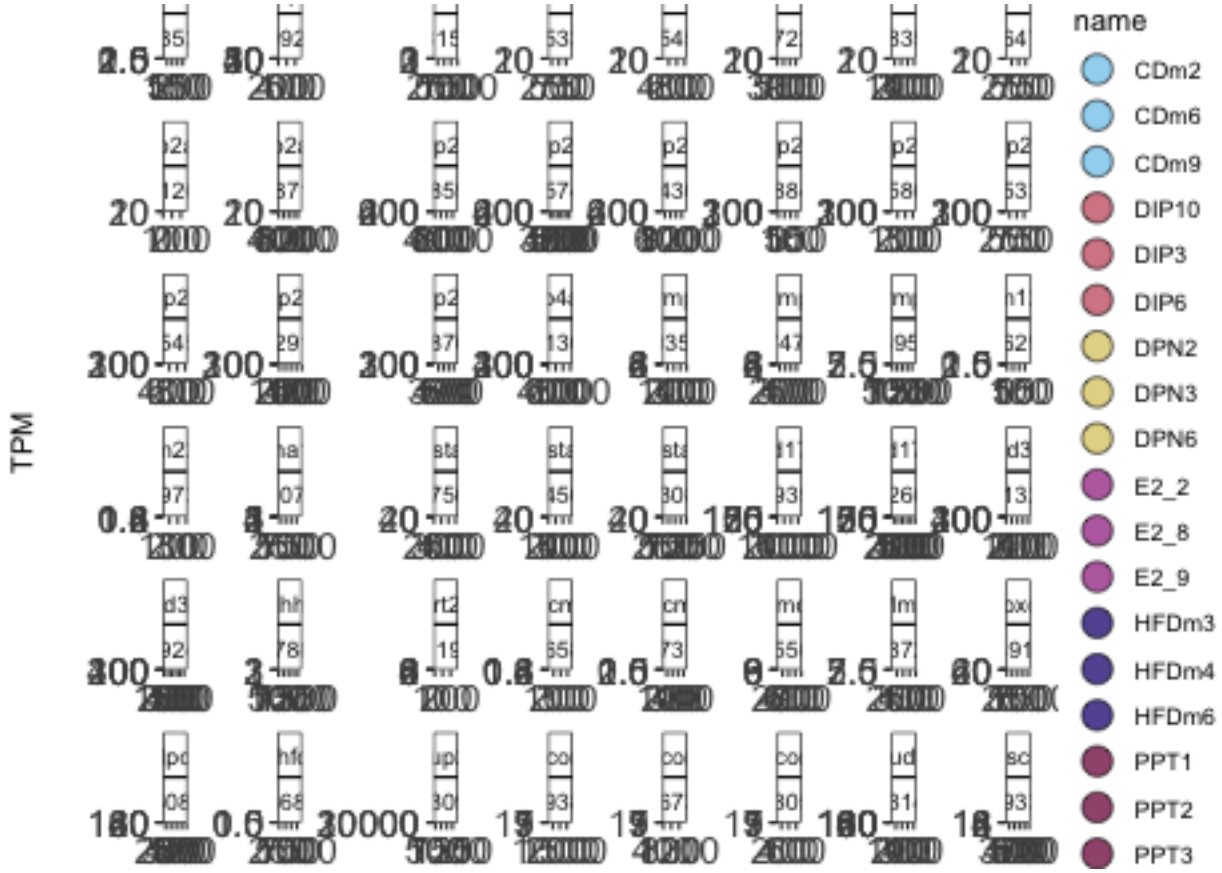
```
## [1] 80
```

```
length(unique(K27_GE_long_group_plot_filt$loc_ID)) # 76 unique enhancers
```

```
## [1] 76
```

```
# 80 total combinations of location IDs and genes, one enhancer goes for two genes.
write.table(K27_GE_long_group_plot_filt, "results/Epigenome_analysis/corr_49genes_76enh_toPlot.txt", ro
```

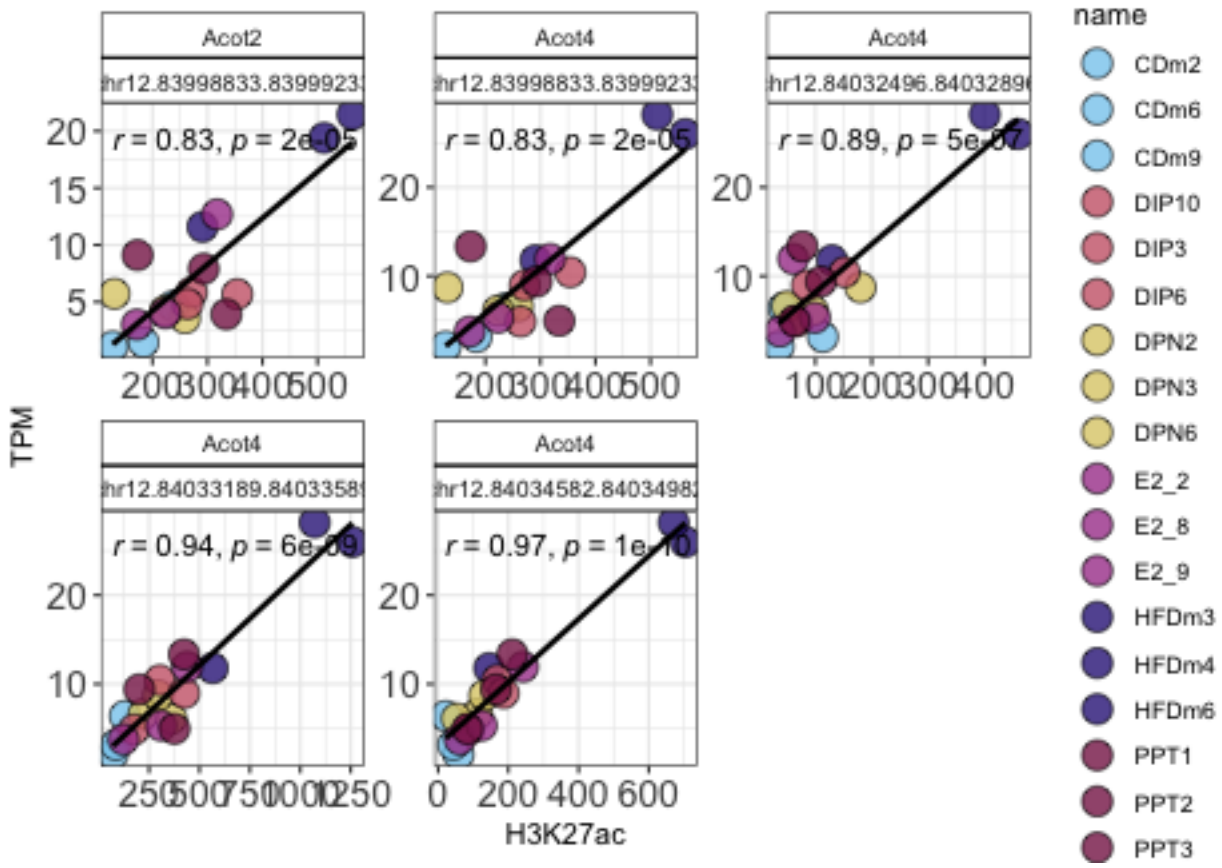
```
library(ggpubr)
  ggscatter(K27_GE_long_group_plot_filt, x = "H3K27ac", y="Gene_expression", add = "reg.line",
            shape=21,size=5, fill="name",alpha=0.8,
            xlab="H3K27ac",
            ylab="TPM",
             palette=c("#88CCEE", "#88CCEE", "#88CCEE",  "#CC6677", "#CC6677","#CC6677", "#DDCC77","#DD
    theme_bw() +
    theme(axis.text = element_text(size=14),
          strip.background = element_rect(colour="black",
                                          fill="white")) +
    facet_wrap(vars(symbol, loc_ID), scales = "free", ncol=8)
```




```
#ggsave("results/Appendix.pdf", width = 50, height=55, units="cm", limitsize=FALSE)

Acot_filt <- K27_GE_long_group_plot_filt %>% dplyr::filter(grepl("Acot", symbol))
ggscatter(Acot_filt, x = "H3K27ac", y="Gene_expression", add = "reg.line",
            shape=21,size=5, fill="name",alpha=0.8,
```

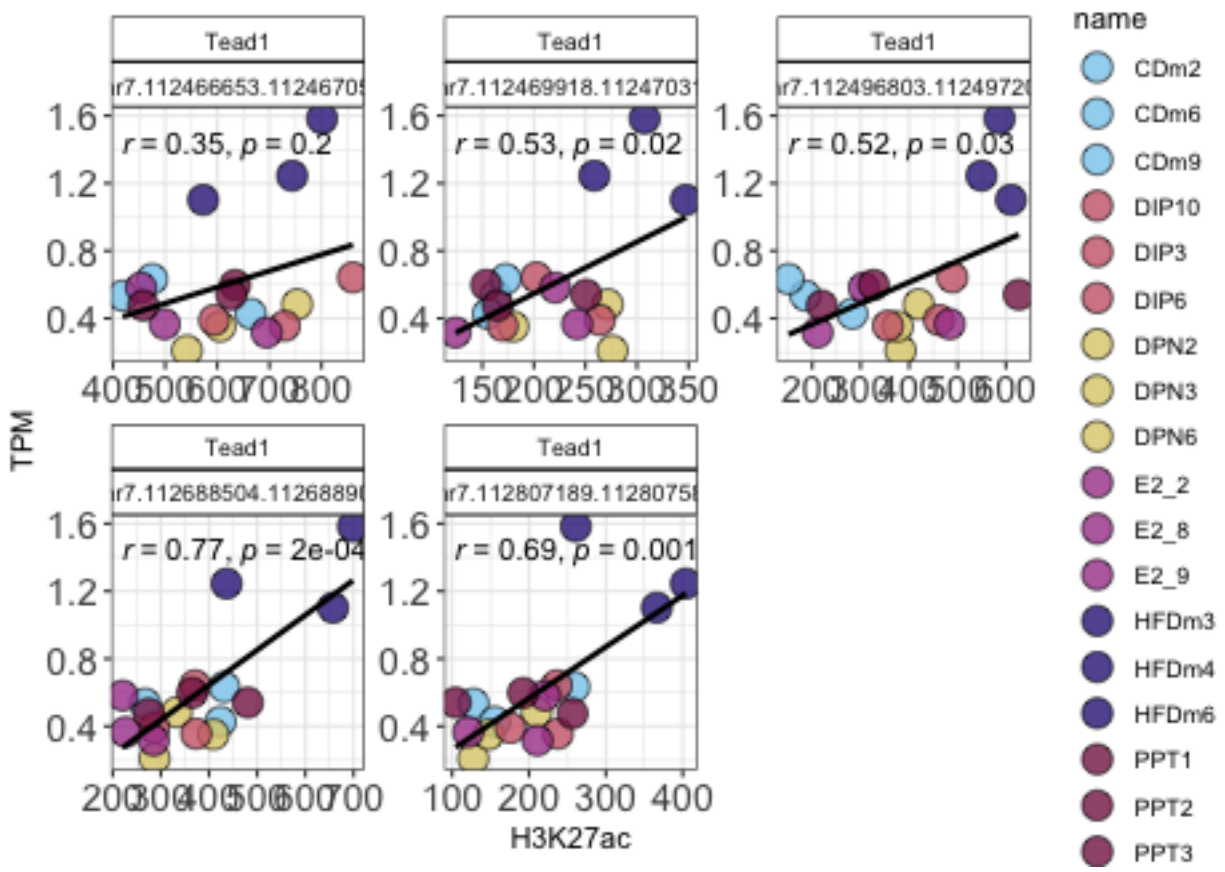
```
            xlab="H3K27ac",
            ylab="TPM",
             palette=c("#88CCEE", "#88CCEE", "#88CCEE",  "#CC6677", "#CC6677","#CC6677", "#DDCC77","#DDC
    stat_cor(method="pearson", p.digits=0, size=4, cor.coef.name = "r") +
    theme_bw() +
    theme(axis.text = element_text(size=14),
          strip.background = element_rect(colour="black",
                                          fill="white")) +
    facet_wrap(vars(symbol, loc_ID), scales="free", ncol=3)
```




```
#ggsave("results/FigEV4F.pdf", width = 20, height=12, units="cm")


# plot Tead1
tead1 <- K27_GE_long_group %>% dplyr::filter(symbol=="Tead1") %>% filter(!loc_ID=="chr7.112688427.11268
# note: this loc_ID is filtered out because it does not have a CTCF peak closeby (at least not a signif
#tead1_filt <- K27_GE_long_group_plot_filt %>% dplyr::filter(symbol=="Tead1")  # Plot this to only show
ggscatter(tead1, x = "H3K27ac", y="Gene_expression", add = "reg.line",
            shape=21,size=5, fill="name",alpha=0.8,
            xlab="H3K27ac",
            ylab="TPM",
            palette=c("#88CCEE", "#88CCEE", "#88CCEE",  "#CC6677", "#CC6677","#CC6677", "#DDCC77","#DDC
    stat_cor(method="pearson", p.digits=0, size=4, cor.coef.name = "r") +
    theme_bw() +
    theme(axis.text = element_text(size=14),
```

```
        strip.background = element_rect(colour="black",
                                        fill="white")) +
    facet_wrap(vars(symbol, loc_ID), scales="free", ncol=3)
```




```
#ggsave("results/Fig4B.pdf", width = 20, height=12, units="cm")
```

## Plot histogram to show in which processes (of the 24 GSEA) the 45 genes fall into.

```
library(clusterProfiler)
K27_GE_long_group_plot_filt <- read.delim("results/Epigenome_analysis/corr_49genes_76enh_toPlot.txt")
length(unique(K27_GE_long_group_plot_filt$symbol))
```

```
## [1] 49
```

```
symbols <- K27_GE_long_group_plot_filt$symbol %>% unique()

reactome_pathways <- readRDS("results/bulkRNAseq_mmus_GSEA_reactome_cluster_sets.rds")
reactome_pathways.2 <- as.data.frame(do.call(cbind, reactome_pathways))

reactome_pathways.long <- pivot_longer(reactome_pathways.2, cols=1:24)
```

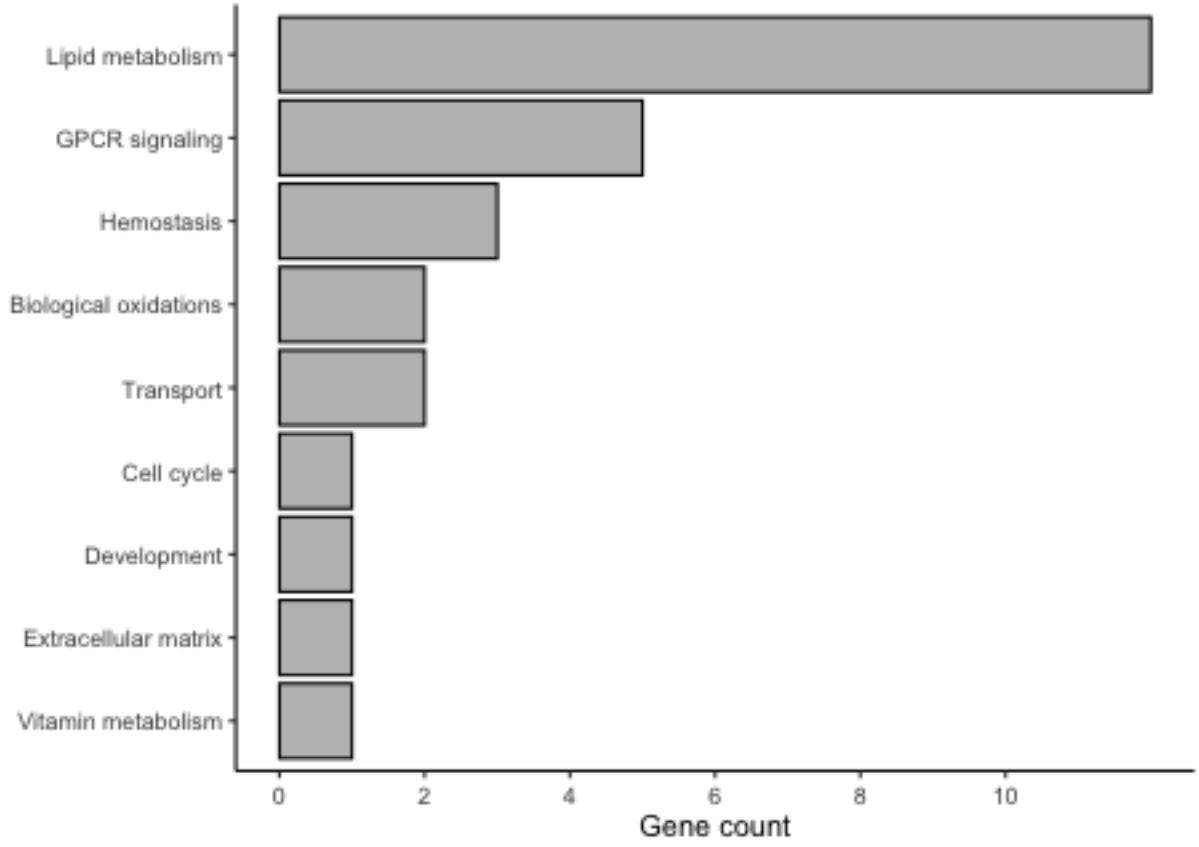
```
intersect_pathways_symbols <- reactome_pathways.long %>% filter(value %in% symbols) %>% unique()
intersect_pathways_symbols_counts <- as.data.frame(table(intersect_pathways_symbols$name))

order.GSEA.pathways <- as.data.frame(table(intersect_pathways_symbols$name)) %>%
  arrange(-Freq) %>%
  dplyr::pull("Var1") %>%
  as.vector()

str(order.GSEA.pathways)
```

```
##  chr [1:9] "Lipid metabolism" "GPCR signaling" "Hemostasis" ...
```

```
ggplot(intersect_pathways_symbols) +
  geom_histogram(aes(x=factor(name, levels=rev(order.GSEA.pathways))), stat="count", fill="grey", color=
  coord_flip() +
  theme_classic() +
  theme(axis.text.x = element_text(vjust = .5)) +
  xlab("") +
  ylab("Gene count") +
  scale_y_continuous(limits = c(), breaks=c(0,2,4,6,8,10))
```




```
library(clusterProfiler)
options(connectionObserver = NULL)
# Warning: call dbDisconnect() when finished working with a connection
```

```r
library(org.Mm.eg.db)

unique_symbols <- K27_GE_long_group_plot_filt$symbol %>% unique()
length(unique_symbols)
```

```
## [1] 49
```

```r
GO_BP <- enrichGO(gene = unique_symbols,
                keyType       = 'SYMBOL',
                OrgDb         = org.Mm.eg.db,
                ont           = "BP",
                pAdjustMethod = "BH",
                 pvalueCutoff  = 0.05,
                qvalueCutoff  = 0.05,
                minGSSize     = 1,
                readable      = F,
                universe = TPM_filt$symbol) # Can also use all expressed genes here instead.
head(as.data.frame(GO_BP))
```

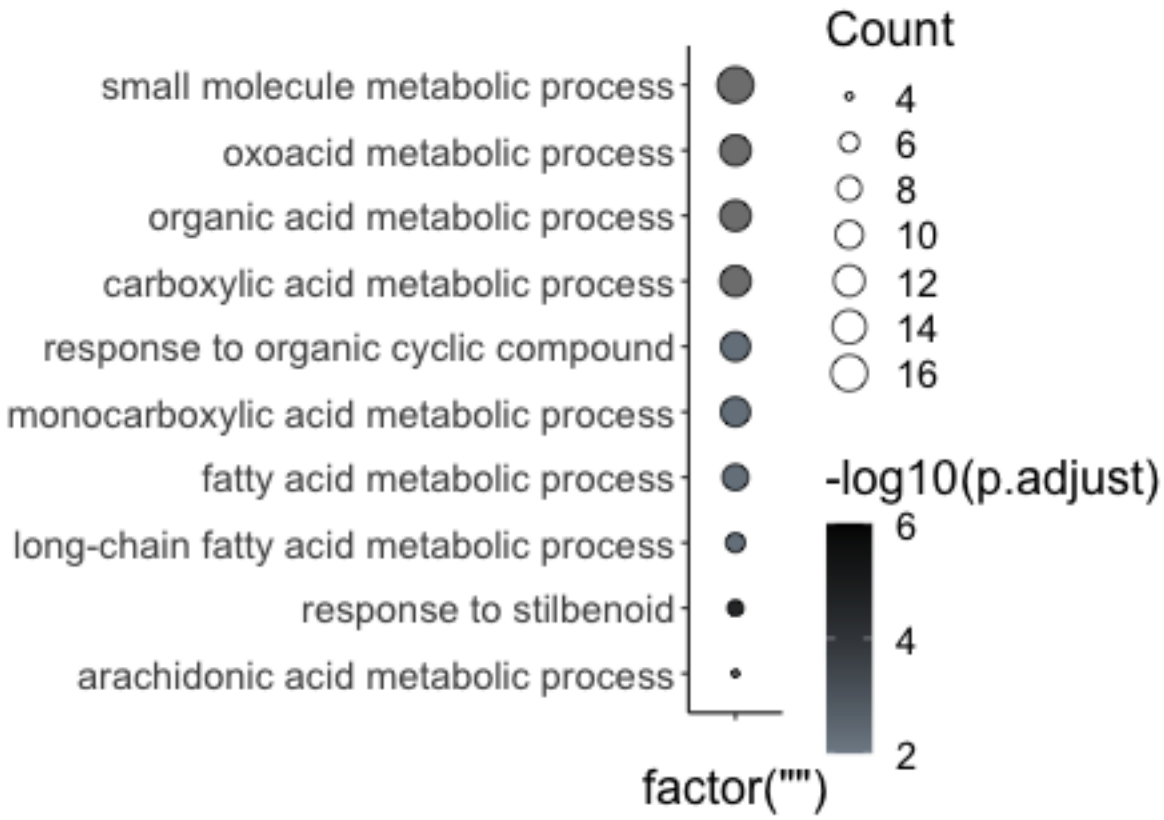
```
##                    ID                          Description GeneRatio
## GO:0035634 GO:0035634                 response to stilbenoid      5/46
## GO:0032787 GO:0032787   monocarboxylic acid metabolic process     11/46
## GO:0001676 GO:0001676 long-chain fatty acid metabolic process      6/46
## GO:0014070 GO:0014070     response to organic cyclic compound     11/46
## GO:0006631 GO:0006631           fatty acid metabolic process      9/46
## GO:0019752 GO:0019752       carboxylic acid metabolic process     12/46
##              BgRatio       pvalue     p.adjust       qvalue
## GO:0035634   7/2986 1.426769e-08 2.568184e-05 2.210741e-05
## GO:0032787 145/2986 7.458397e-06 5.230756e-03 4.502732e-03
## GO:0001676  34/2986 9.299458e-06 5.230756e-03 4.502732e-03
## GO:0014070 153/2986 1.257135e-05 5.230756e-03 4.502732e-03
## GO:0006631 100/2986 1.452988e-05 5.230756e-03 4.502732e-03
## GO:0019752 213/2986 5.662355e-05 1.463852e-02 1.260112e-02
##                                                                                  geneID
## GO:0035634                                                    Hsd3b5/Cyp2b9/Cyp2a5/Cd36/Gsta2
## GO:0032787        Mthfd1l/Aldh3a2/Acot2/Cyp2b9/Cyp2a5/Cyp2a22/Acot4/Mpc1/Cyp4a10/Cd36/Nudt7
## GO:0001676                                          Cyp2b9/Cyp2a5/Cyp2a22/Acot4/Cyp4a10/Cd36
## GO:0014070                Gna14/Stk39/Bche/Hsd3b5/Ncor2/Cyp2b9/Cyp2a5/Cdh1/Cd36/Gsta2/Sstr2
## GO:0006631                     Aldh3a2/Acot2/Cyp2b9/Cyp2a5/Cyp2a22/Acot4/Cyp4a10/Cd36/Nudt7
## GO:0019752 Mthfd1l/Aldh3a2/Acot2/Slc7a7/Cyp2b9/Cyp2a5/Cyp2a22/Acot4/Mpc1/Cyp4a10/Cd36/Nudt7
##            Count
## GO:0035634     5
## GO:0032787    11
## GO:0001676     6
## GO:0014070    11
## GO:0006631     9
## GO:0019752    12
```

```r
View(as.data.frame(GO_BP))

write.table(as.data.frame(GO_BP), "Supplementary_tables/SupplementaryTable_GO_BP_ESEG_genes.txt", row.na
```

```r
library(dplyr)
library(ggplot2)
  plot_me_ordered <- GO_BP[order(GO_BP$p.adjust), ]
  plot_me_ordered <- plot_me_ordered[1:10, ]
  plot_me_ordered <- plot_me_ordered[order(plot_me_ordered$Count), ]
  name_order <- plot_me_ordered %>%
    dplyr::pull("Description")

  ggplot(plot_me_ordered, aes(x=factor(Description, levels=name_order), fill=-log10(p.adjust), y=factor
    geom_point(shape=21, aes(size=Count, fill=-log10(p.adjust))) +
    coord_flip() +
    scale_fill_gradient(low = "#808b96", high = "black",
    limits = c(2, 6), breaks = c(2, 4, 6))+
    theme_classic() +
    theme(text=element_text(size = 18)) +
    ggtitle("") +
    xlab("")
```




```r
sessionInfo()
```

```
## R version 4.2.3 (2023-03-15)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
```

```
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] clusterProfiler_4.6.2 ggpubr_0.6.0          biomaRt_2.54.1
##  [4] org.Mm.eg.db_3.16.0   AnnotationDbi_1.60.2  Biobase_2.58.0
##  [7] ChIPpeakAnno_3.32.0   GenomicRanges_1.50.2  GenomeInfoDb_1.34.9
## [10] IRanges_2.32.0        S4Vectors_0.36.2      BiocGenerics_0.44.0
## [13] lubridate_1.9.2       forcats_1.0.0         stringr_1.5.0
## [16] dplyr_1.1.2           purrr_1.0.2           readr_2.1.4
## [19] tidyr_1.3.0           tibble_3.2.1          ggplot2_3.4.3
## [22] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##   [1] utf8_1.2.3                  tidyselect_1.2.0
##   [3] RSQLite_2.3.1               grid_4.2.3
##   [5] BiocParallel_1.32.6         scatterpie_0.2.1
##   [7] munsell_0.5.0               codetools_0.2-19
##   [9] withr_2.5.0                 colorspace_2.1-0
##  [11] GOSemSim_2.24.0             filelock_1.0.2
##  [13] highr_0.10                  knitr_1.43
##  [15] rstudioapi_0.15.0           ggsignif_0.6.4
##  [17] DOSE_3.24.2                 MatrixGenerics_1.10.0
##  [19] labeling_0.4.2              GenomeInfoDbData_1.2.9
##  [21] polyclip_1.10-4             bit64_4.0.5
##  [23] farver_2.1.1                downloader_0.4
##  [25] vctrs_0.6.3                 treeio_1.22.0
##  [27] generics_0.1.3              gson_0.1.0
##  [29] lambda.r_1.2.4              xfun_0.39
##  [31] timechange_0.2.0            BiocFileCache_2.6.1
##  [33] regioneR_1.30.0             R6_2.5.1
##  [35] graphlayouts_1.0.0          AnnotationFilter_1.22.0
##  [37] bitops_1.0-7                cachem_1.0.8
##  [39] fgsea_1.24.0                gridGraphics_0.5-1
##  [41] DelayedArray_0.24.0         BiocIO_1.8.0
##  [43] scales_1.2.1                ggraph_2.1.0
##  [45] enrichplot_1.18.4           gtable_0.3.3
##  [47] tidygraph_1.2.3             ensembldb_2.22.0
##  [49] rlang_1.1.1                 splines_4.2.3
##  [51] rtracklayer_1.58.0          rstatix_0.7.2
##  [53] lazyeval_0.2.2              broom_1.0.5
##  [55] yaml_2.3.7                  reshape2_1.4.4
##  [57] abind_1.4-5                 GenomicFeatures_1.50.4
##  [59] backports_1.4.1             qvalue_2.30.0
##  [61] RBGL_1.74.0                 tools_4.2.3
##  [63] ggplotify_0.1.1             RColorBrewer_1.1-3
##  [65] Rcpp_1.0.11                 plyr_1.8.8
```

```
##  [67] progress_1.2.2              zlibbioc_1.44.0
##  [69] RCurl_1.98-1.12             prettyunits_1.1.1
##  [71] viridis_0.6.3               cowplot_1.1.1
##  [73] SummarizedExperiment_1.28.0 ggrepel_0.9.3
##  [75] magrittr_2.0.3              data.table_1.14.8
##  [77] futile.options_1.0.1        ProtGenerics_1.30.0
##  [79] matrixStats_1.0.0           hms_1.1.3
##  [81] patchwork_1.1.3             evaluate_0.21
##  [83] HDO.db_0.99.1               XML_3.99-0.14
##  [85] VennDiagram_1.7.3           gridExtra_2.3
##  [87] compiler_4.2.3              crayon_1.5.2
##  [89] shadowtext_0.1.2            htmltools_0.5.5
##  [91] ggfun_0.1.1                 mgcv_1.8-42
##  [93] tzdb_0.4.0                  aplot_0.1.10
##  [95] DBI_1.1.3                   tweenr_2.0.2
##  [97] formatR_1.14                dbplyr_2.3.3
##  [99] MASS_7.3-58.2               rappdirs_0.3.3
## [101] Matrix_1.5-3                car_3.1-2
## [103] cli_3.6.1                   parallel_4.2.3
## [105] igraph_1.5.0                pkgconfig_2.0.3
## [107] GenomicAlignments_1.34.1    xml2_1.3.5
## [109] InteractionSet_1.26.1       ggtree_3.6.2
## [111] multtest_2.54.0             XVector_0.38.0
## [113] yulab.utils_0.0.6           digest_0.6.33
## [115] graph_1.76.0                Biostrings_2.66.0
## [117] rmarkdown_2.23              fastmatch_1.1-3
## [119] tidytree_0.4.4              restfulr_0.0.15
## [121] curl_5.0.1                  Rsamtools_2.14.0
## [123] rjson_0.2.21                lifecycle_1.0.3
## [125] nlme_3.1-162                jsonlite_1.8.7
## [127] carData_3.0-5               futile.logger_1.4.3
## [129] viridisLite_0.4.2           BSgenome_1.66.3
## [131] fansi_1.0.4                 pillar_1.9.0
## [133] lattice_0.20-45             KEGGREST_1.38.0
## [135] fastmap_1.1.1               httr_1.4.6
## [137] survival_3.5-3              GO.db_3.16.0
## [139] glue_1.6.2                  png_0.1-8
## [141] bit_4.0.5                   ggforce_0.4.1
## [143] stringi_1.7.12              blob_1.2.4
## [145] memoise_2.0.1               ape_5.7-1
```